

**TextGrid – das Community-Grid für die
Geisteswissenschaften**



www.TextGrid.de

**Die TextGrid-Infrastruktur
Informationen für Kooperationspartner**

Version 1.0

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Überblick.....	3
Das Projekt	3
Motivation	3
Kollaborativer Ansatz.....	4
Gestaltungsprinzipien	4
TextGridLab – der Werkzeugkasten	4
Die Benutzeroberfläche.....	4
Die TextGrid-Tools	5
TextGridRep – Datenanbindung und Langzeitarchivierung	7
Anwendungsmöglichkeiten	8
Kooperationsszenarien	8
Anhang: Technische Rahmenbedingungen.....	10
Überblick	10
GUI.....	10
Tools / Services.....	10
Middleware und Grid-Anbindung.....	11

Überblick

Der folgende Text ist für potentielle Kooperationspartner gedacht, die die vom TextGrid-Projekt entwickelte Infrastruktur nutzen bzw. ausbauen wollen. Er soll einen kurzen Überblick über das Projekt und die Komponenten der TextGrid-Software und -Infrastruktur vermitteln, um abschließend mögliche Kooperationszenarien aufzuzeigen. Technische Hintergrundinformationen finden sich im Anhang. Dieser Text dient der Vorbereitung möglicher Kooperationen. Konkrete Ziele einer Kooperation sowie die Verantwortlichkeiten in der Zusammenarbeit müssen im Einzelnen diskutiert werden.

Das Projekt

TextGrid¹ entwickelt eine Forschungsinfrastruktur für die Geisteswissenschaften auf Basis von Grid-Technologien – ein Ansatz, der bis dato in den Geisteswissenschaften in Deutschland nicht verfolgt wurde, aber im internationalen Kontext im Rahmen von eScience- und eResearch-Initiativen und -Projekten bereits sichtbar ist.

TextGrid beteiligt sich aktiv am Aufbau einer eHumanities-Infrastruktur in Deutschland. Ein im Januar 2008 gestartetes DFG-Projekt entwickelt eine Gesamt-Konzeption für die eHumanities-Initiative, während BMBF-Diskussionen darauf abzielen, eine langfristige Roadmap zu entwickeln (Agenda 2020 für eHumanities in Deutschland – Fahrplan für die Etablierung einer virtuellen Forschungsumgebung für die Humanities).

Als einziges geisteswissenschaftliches Community-Projekt in D-Grid² bringt TextGrid die speziellen Anforderungen und Sichtweisen aus den Geisteswissenschaften in D-Grid ein. Während das World Wide Web in der Regel noch immer nur statische Informationen bereitstellt, ermöglicht das Grid nun den direkten Zugriff auf die Ressourcen selbst, wie Rechner, Speicher, wissenschaftliche Instrumente und Experimente, Anwendungen und Daten. Grid-Infrastrukturen bieten den Wissenschaftlern eine Vielzahl von Vorteilen, wie zum Beispiel den einheitlichen Zugriff auf verteilte Ressourcen, nahezu unbegrenzte Rechen- und Speicherkapazität sowie hohe Flexibilität durch den dynamischen und konzertierten Betrieb der vernetzten Ressourcen. "Ressourcen" sind in den Geisteswissenschaften vor allem Daten und Dienste (Anwendungen) – bei rechenintensiven Prozessen wie z.B. Migration im Rahmen der Langzeitarchivierung oder OCR von großen Datenmengen wird jedoch zunehmend auch virtualisierte Hardware als Ressource für diesen Wissenschaftsbereich bedeutsam. Dabei sind aber die Langzeitarchivierung der Datenressourcen, die Verwaltung von Wissen und die Bearbeitung und Verarbeitung von Textdaten für die Geisteswissenschaften von fundamentaler Bedeutung.

Motivation

Wenn Textwissenschaftler das Beziehungsgeflecht zwischen Sprache und Diskurs oder die komplexen Prozesse des Entstehens literarischer Texte untersuchen, arbeiten sie oft noch vereinzelt oder innerhalb in sich geschlossener Projekte. Aktuelle Forschungsaktivitäten im Bereich der Textwissenschaften lassen darüber hinaus die Verknüpfung mit existierenden Textkorpora, Wörterbüchern, Lexika oder Sekundärliteratur und die Anbindung an bereits verfügbare Textwerkzeuge vermissen, obwohl dies einen beträchtlichen Mehrwert darstellen und eine Vielzahl an Möglichkeiten der weiteren Datenverarbeitung schaffen würde. TextGrid stellt mit seiner Architektur und seinen Werkzeugen eine

¹ Gefördert vom Bundesministerium für Bildung und Forschung (BMBF) im Rahmen der D-Grid Initiative, Förderkennzeichen 07TG01A-H. Weitere Informationen zu TextGrid unter <http://www.textgrid.de>.

² Die deutsche Grid-Initiative, <http://www.d-grid.de>.

Infrastruktur zur Verfügung, die eine solche Integration möglich macht und die deshalb die Arbeitsweise von Geisteswissenschaftlern nachhaltig verändern kann. Die Notwendigkeit, eine Forschungsinfrastruktur mit gridfähiger Architektur auch für die Geisteswissenschaften zu schaffen, liegt auf der Hand: Durch die Digitalisierungsinitiativen der letzten Jahre sind erhebliche Datenmengen entstanden, die inzwischen einige hundert Terabyte umfassen – Grids können mit solchen großen, ständig wachsenden Datenvolumen umgehen³.

Kollaborativer Ansatz

TextGrid errichtet eine modulare, gridbasierte Infrastruktur, die als Grundlage für kollaborative virtuelle Arbeitsumgebungen dient. Grid-Strukturen erlauben es, Wissenschaftler miteinander zu vernetzen, die derzeit in räumlicher Distanz an vergleichbaren Projekten arbeiten, sowie vorhandene Tools global verfügbar zu machen, die bislang lediglich lokal benutzbar sind. Auf diese Weise wird eine Plattform errichtet, auf der viele Experten verschiedener Fachgebiete eine virtuelle Arbeitsgemeinschaft bilden können. Während der Projektphase geschieht dies mit einem Schwerpunkt auf der germanistischen Editionsphilologie – und ansatzweise für die Linguistik. So ermöglicht und unterstützt TextGrid derzeit die gemeinschaftliche philologische Bearbeitung, Analyse, Annotation, Edition und Publikation von Textdaten.

Gestaltungsprinzipien

Bei der Konzeption der TextGrid-Architektur standen drei Aspekte im Vordergrund:

1. Die Bedienung soll so einfach und intuitiv wie möglich sein.
2. Es muss mit einfachen Mitteln möglich sein, TextGrid den individuellen Bedürfnissen anzupassen.
3. TextGrid soll mit bestehenden Infrastrukturen und Werkzeugen zusammenarbeiten können.

Um der ersten Anforderung zu genügen, wurde und wird bei der Konzeption und Gestaltung der Benutzeroberfläche ein hoher Aufwand betrieben. Für 2008 sind ausführliche Testreihen unter Beteiligung zahlreicher Fachwissenschaftler aus unterschiedlichen Disziplinen geplant. Darüber hinaus wurde versucht, die Komplexität des Grids in den tieferliegenden Architekturschichten zu kapseln. Für die Nutzer ist es in der Regel von wenig Interesse, auf welchen Servern die Daten liegen, wohin sie aus Gründen der Ausfallsicherheit und Performance repliziert werden, solange sie – unter Berücksichtigung der jeweils festgelegten Zugriffsrechte – zuverlässig verfügbar und durchsuchbar sind. Um dem zweiten und dritten Kriterium zu genügen, setzt TextGrid auf offene Standards und ein modulares System verteilter, plattformunabhängiger *Open-Source*-Komponenten. Mehr dazu im Anhang.

TextGridLab – der Werkzeugkasten

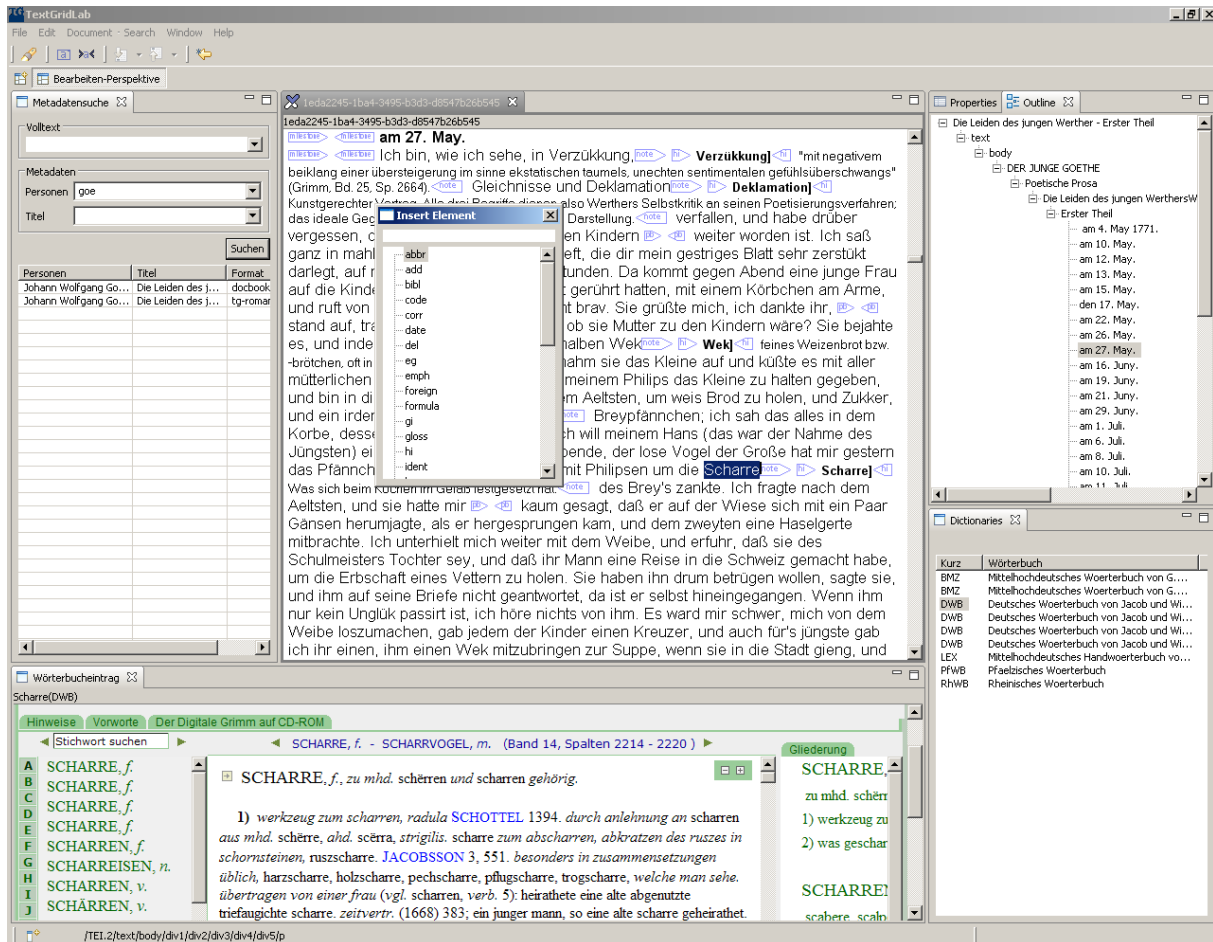
Das *TextGridLab* (Laboratory) bietet dem Nutzer einen integrierten Zugang zu allen Werkzeugen in TextGrid (s.u.). Es basiert auf einem offenen Netzwerk von *Open-Source*-Tools, das für neue Nutzergruppen einfach erweiterbar und anpassbar ist.

Die Benutzeroberfläche

Die grafische Benutzeroberfläche von TextGrid basiert auf Eclipse, einer *Open-Source*-Entwicklungsumgebung, die u.a. zur Erstellung grafischer Oberflächen benutzt wird. An der Entwicklung sind Firmen wie Borland, IBM, Intel, Oracle und SAP beteiligt.

³ Wichtige Kriterien hierbei sind: Ausfallsicherheit, hohe Verfügbarkeit und schneller Zugriff.

Zentrales Element ist der XML-Editor, um den sich die anderen Module wie der Workflow-Editor, die Projektverwaltung und Rechtevergabe, Suchmasken sowie die Bedienelemente der einzelnen Werkzeuge gruppieren. Die Gruppierung und Auswahl der Elemente lässt sich entsprechend den individuellen (Projekt-)Anforderungen konfigurieren.



Prototyp der Benutzeroberfläche (von links nach rechts: Suchmasken, XML-Editor, Struktursicht, Schnittstelle zum Trierer Wörterbuchnetz)

Die TextGrid-Tools

Als Grundausstattung werden in der Projektphase folgende Tools⁴ entwickelt:

Interaktive Tools

XML-Editor – Der Editor erlaubt das Bearbeiten von XML-Dateien lokal oder im Grid. Der Anwender kann dabei beliebig wechseln zwischen einer eher technisch orientierten Ansicht, die alle Tags und Attribute anzeigt, und einer Ansicht, die eher an der Darstellung eines Textverarbeitungsprogramms orientiert ist.

⁴ Für eine ausführlichere Funktionsbeschreibungen siehe die beiden Reports 2.1 und 2.2 unter http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_2_1.pdf und http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid-R2.2_ToolsII.pdf.

Link-Editor Text (Text ↔ Text) – Der Link-Editor Text ist eine Eingabehilfe für Links in XML-Dateien und verbindet Elemente von Recherchetooll und XML-Editor. Benutzer haben die Möglichkeit, Links zu beliebigen Elementen in TextGrid-Dokumenten in der aktuellen (TEI-)Datei zu generieren.

Grafischer Link-Editor (Text ↔ Bild) – Der grafische Linkeditor hat die Aufgabe, den XML-Editor bei der Alignierung von Text und Bildelementen zu unterstützen. Ziel ist die Erstellung einer Ausgabedatei, die die Textelemente und die topographische Beschreibung enthält.

Workflow-Editor – Der Workflow-Editor erlaubt es, die Automatisierung von Arbeitsabläufen (Workflows) zu definieren (orchestrieren).

Recherchetooll – Das Recherchetooll dient der Recherche und dem Retrieval von Struktur- und Metadaten. Eine semantische Suche (RDF-basiert) ist ebenfalls möglich.

Kollationierung – Zwei oder mehr XML (bzw. TEI⁵)-kodierte Dokumente werden verglichen und ihre Unterschiede werden nach TEI kodiert und notiert. Bei diesem Tool ist auch ein Betrieb auch im Batch-Modus möglich (→ Streaming Tool).

Projektverwaltung (Datei- und Rechtemanagement) – Die Projektverwaltung ermöglicht die Erstellung neuer Projekte durch den Projektleiter und die Zuordnung von weiteren Benutzern (in bestimmten Rollen) zu einem Projekt, sowie die Abfrage und das Setzen von Rechten an TextGrid-Objekten.

Projektbrowser / Navigationstool – Der Projektbrowser ist ein ständiger Begleiter im TextGridLab. Es ermöglicht den einfachen Zugriff auf alle Projektmaterialien eines Benutzers. Der Navigator stellt damit einen möglichen Startpunkt für den Nutzer im TextGridLab dar.

Metadatenannotation – Ein generisches Werkzeug, das dazu dient, strukturierte Daten über eine definierbare Maske zu erfassen und an einer frei bestimmbar Position in einer Datei einzufügen / abzuspeichern. Anwendungsfälle sind z.B. die Eingabe von Metadaten zu einem TextGrid-Objekt (Text, Bild etc.) oder die allgemeine Eingabe strukturierter Daten in eine Datei bzw. an einer bestimmten Stelle in einer Datei.

Bibliographietool – Das Bibliographietool dient dazu, bibliographische Daten entweder aus bereits vorhandenen Datenbeständen zu importieren, insbes. aus Bibliothekskatalogen (z.B. zvdd, Kalliope) und den TEI-Headern (Metadaten) bestehender (TextGrid-)Objekte, Bibliographien über eine definierbare Maske zu erfassen, zu bearbeiten, zu verwalten und an einer frei bestimmbar Position in einer Datei einzufügen / abzuspeichern bzw. bibliographische Daten in bestimmten Standard-Formaten zu exportieren (z.B. TEI, MODS).

Streaming Tools

Hierbei handelt es sich um nicht-interaktive Tools, die als Batchprozesse gestartet werden. Üblicherweise verfügen sie jedoch über eine interaktive GUI-Komponente zur Konfiguration.

Tokenizer – Zerlegt einen Text in eine Folge logischer Einheiten (Tokens), hier: Wörter und Satzzeichen. Diese werden durch Anfangs- und Endemarkierungen gekennzeichnet. Die hierbei zu verwendenden Elemente lassen sich ebenso wie vordefinierte Tokens wie Abkürzungen, Eigennamen oder reguläre Ausdrücke (z.B. für Datumsangaben) in der Tool-Konfiguration definieren.

⁵ TEI = Text Encoding Initiative (<http://www.tei-c.org>), ein XML-Standard zur Auszeichnung von (Quell-)Texten.

Lemmatizer (morphologische Analyse) – Analysiert einzelne Wortformen (Tokens) und liefert als Ausgabe a) das zugehörige Lemma, d.h. das Token wird auf seine grammatische Grundform zurückgeführt, b) die zugehörige Wortart (PartOfSpeech) und c) weitere morphologische Merkmale (Numerus, Genus etc.).

Sortiertool – Mit Hilfe des Sortiertools kann ein Anwender gegebene Folgen von Zeichenketten gemäß kulturellen und fachlichen Erwartungen anordnen. Die dabei zu berücksichtigenden Sortierschlüssel und Sortierkonventionen kann der Nutzer frei vorgeben. Zur Vereinfachung der Handhabung erlaubt das Sortiertool aber auch, sich direkt auf einschlägige nationale und europäische Standards zu beziehen.

Streaming Editor – Transformationen von Dateien aufgrund von Regeln, z.B. automatisierte Anreicherung potentiell unstrukturierter Texte mit XML-Strukturen. Die Eingabe muss nicht XML sein, sondern kann ein beliebiges Textformat sein (etwa OCR-Rohdaten, reiner Text), die Ausgabe kann, muss aber nicht XML sein.

Text Publisher Web – Der Webpublisher dient Projekten dazu, ihre Daten und Ergebnisse im Rahmen eines Internetauftritts zu präsentieren. Hierfür stellt TextGrid vorgefertigte Komponenten und die Schnittstelle zur Datenanbindung an das TextGridRep (s.u.) zur Verfügung.

[Text Publisher Print] – Aufgabe des Print Publishers ist der Satz komplexer XML-Daten, der den Anforderungen an wissenschaftliche Publikationen genügt. Anders als in existierenden Satzprogrammen mit grafischer Benutzerschnittstelle soll der Anwender direkt mit den semantisch annotierten Daten arbeiten, ohne diese erst in ein programmspezifisches Layoutsystem überführen zu müssen. Dieses Tool soll allen textproduzierenden Wissenschaftlern die einfache Gestaltung ihrer Daten und Ausgabe nach PDF ermöglichen, sei es für den Druck oder für die Langzeitarchivierung. Die Entwicklung wird separat beantragt⁶.

TextGridRep – Datenanbindung und Langzeitarchivierung

Das *TextGridRep* (Repository) ermöglicht die strukturierte Speicherung und langfristige Aufbewahrung von wissenschaftlichen Texten im Grid. Die Texte werden bei der Registrierung automatisch mit Metadaten verknüpft und semantisch indiziert.

Die im Grid liegenden Daten lassen sich vom TextGridLab aus bearbeiten, verwalten, Projekten zuordnen, mit Zugriffsrechten und Metadaten versehen. Ein Projektbrowser dient der Navigation über Projekte und die im Grid liegenden Objekte (Texte, Abbildungen etc.). Der Objektzugriff erfolgt über die Metadaten, bei der Suche zusätzlich über die Textdaten (Volltext- und strukturierte Suche⁷).

Um in TextGrid auf Daten zugreifen zu können, müssen diese aber nicht notwendigerweise im Grid liegen. Für die Anbindung und Integration von Daten sind verschiedene Stufen denkbar: von der Bereitstellung von Schnittstellen für Recherche und Text-Retrieval⁸, über den Import der Metadaten, bis hin zur vollständigen Integration der Daten in die TextGrid-Infrastruktur.

⁶ Es hat sich gezeigt, dass die Anforderungen an ein derartiges Tool das, was im Rahmen von TextGrid geleistet werden kann, bei weitem sprengt. Die Projektpartner an der FH Worms und an der Universität Trier haben deshalb zusammen mit Saphor einen DFG-Antrag für ein separates Vorhaben eingereicht, in dessen Rahmen eine derartige Print-Komponente entwickelt werden soll.

⁷ Z.B. bei XML-Daten nach Textsorten-spezifischem Markup (zum Konzept der TextGrid-Kerncodierung siehe http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Kerncodierung_070615.pdf).

⁸ Als Beispiel wäre hier die im Projektrahmen erfolgte Anbindung des Trierer Wörterbuchnetzes (<http://www.woerterbuchnetz.de>) über eine Web Service-Schnittstelle zu nennen.

TextGrid bietet außerdem die Möglichkeit zur Langzeitarchivierung. Entsprechende *Policies* werden derzeit entwickelt.

Anwendungsmöglichkeiten

Projekte können die vorhandenen Dienste nach ihren Bedürfnissen zu standardisierten Arbeitsabläufen⁹ kombinieren und auch die bereitgestellten Programme ihren eigenen Bedürfnissen anpassen. Aufwendigere Anpassungen könnten auch für andere Projekte von Interesse sein – dies gilt insbesondere für zusätzliche Funktionalitäten, die im Kontext von Projekten entwickelt werden oder gar für die Entwicklung eigener Dienste oder Module. TextGrid bemüht sich, all diese Formen der Verwendung durch eine ausführliche Dokumentation und einfach zu übernehmende Beispiele zu unterstützen. Außerdem wird es einschlägige Informations- und Diskussionsmöglichkeiten für TextGrid-Benutzer geben (Portal, Forum).

Kooperationsszenarien

Im Folgenden werden exemplarisch einige mögliche Nutzungs- bzw. Kooperationsszenarien dargestellt. Diese finden auf verschiedenen Ebenen der TextGrid-Gesamtarchitektur statt, die in der Abbildung unten dargestellt wird. Hierbei steht die dritte Säule für weitere Zweige der Geisteswissenschaften, wie z.B. weitere Philologien, Geschichte, Kulturwissenschaften etc.

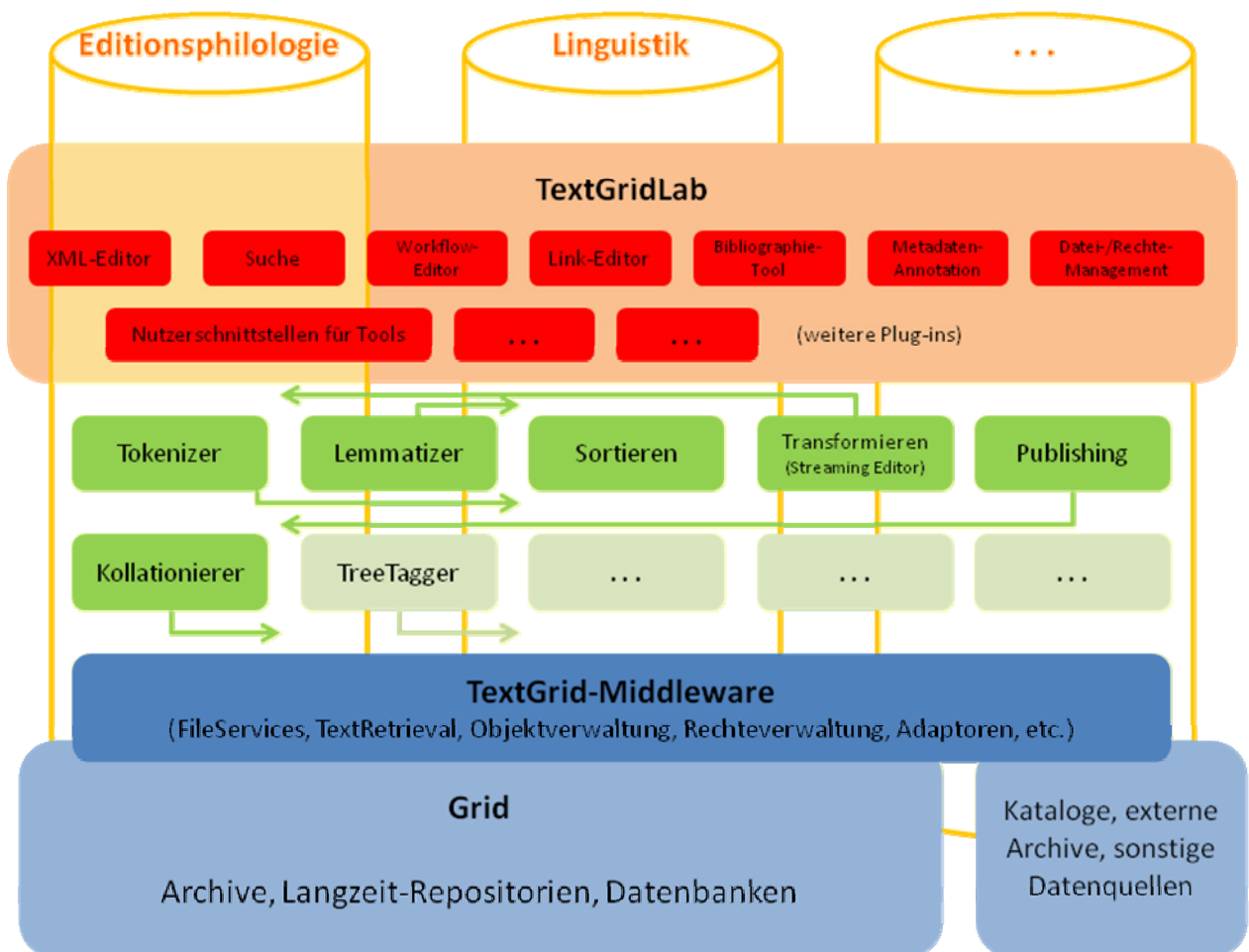
1. Ein Projekt möchte die existierende Software-Infrastruktur nutzen. Hierzu muss die GUI des *TextGridLab* (rot) lokal installiert sein. Danach kann man auf bereits vorkonfigurierte Dienste im Netz (grün) zugreifen und eigene lokal gespeicherte Dateien damit bearbeiten. Darüber hinaus ist über die TextGrid-Middleware (blau) auch lesender Zugriff auf öffentlich zugängliche Datei-Objekte möglich.
2. Möchte ein Projekt zusätzlich auf die Datei-Dienste der TextGrid-Middleware (blau) zugreifen, also z.B. Dateien im Grid speichern, lesen, verändern oder löschen sowie Metadaten zu den eigenen Dateien in die zentrale Metadatenbank einbringen, müssen sich die Projektmitglieder authentifizieren (entweder über den eigenen Account der Heimatorganisation, oder über einen bei TextGrid zu beantragenden Account) und ein TextGrid-Projekt anlegen, unter welchem die Dateien organisiert werden. Ein Projektleiter kann weitere Personen in das Projekt aufnehmen und ihnen Rollen – und damit entsprechende Zugriffsrechte zuordnen.
3. Hat ein Projekt ein eigenes Tool erstellt, kann es dieses in die TextGrid-Infrastruktur integrieren. Muss dieses Tool nicht auf die TextGrid-Middleware zugreifen, reicht es aus, eine Web Services-Schnittstelle zur Verfügung zu stellen, über die Ein- und Ausgabedateien und Konfigurationsparameter über das Netz transportiert werden können (bleibt im grünen Bereich). Soll das Tool direkt über die GUI (und nicht nur indirekt über den Workflow-Editor) angesprochen werden, muss zusätzlich ein Eclipse-Plugin (rot) entwickelt und integriert werden. Sollen auch Middleware-Dienste (blau) vom Tool in Anspruch genommen werden, müssen TextGrid-spezifische Parameter (Authentifizierungs- und Logging-Information) in den Web Service eingebaut werden.
4. Möchte ein Projekt ein bestehendes Archiv von Daten in das *TextGridRep* einbringen, so kann dies über ein Bulkupload-Tool oder über den Workflow-Editor realisiert werden. Hierbei

⁹ Einen Überblick über die in TextGrid zunächst unterstützten Arbeitsabläufe bietet http://www.textgrid.de/fileadmin/TextGrid/TextGrid-Szenarien_061212.pdf.

müssen die einzubringenden Metadaten zur Verfügung gestellt werden – u.U. ist hier individueller Programmieraufwand erforderlich, z.B. zur automatischen Metadaten-Extraktion bei Bilddateien oder zum Auslesen von (Meta-)Daten aus bestehenden Datenbanken. Wie oben erwähnt sind verschiedene Stufen der Daten-Anbindung (Metadaten, Schnittstelle zu externen Daten) an die TextGrid-Infrastruktur möglich. Allerdings eröffnet erst die Integration in *TextGridRep* die gesamte Bandbreite der Bearbeitungs- und Retrieval-Möglichkeiten.

Grundsätzlich steht das TextGrid-Projekt-Team gerne bereit, bei notwendigen Integrationsarbeiten mit Rat zur Seite zu stehen bzw. Entwicklungsaufträge anzunehmen.

Für weitere Informationen siehe v.a. die Berichte und Publikationen unter <http://www.textgrid.de>. Rückfragen per E-Mail bitte an info@textgrid.de.



Anhang: Technische Rahmenbedingungen

Überblick

Fachdisziplinen und Projekte können vorhandene Module zu spezifischen Workflows kombinieren und die existierenden Tools durch eigene Programme ergänzen. Das Gleiche gilt auch für die Komponenten der Benutzeroberfläche. Eigene Datenbestände lassen sich bei Bedarf vollständig in die gridbasierte Storage-Infrastruktur integrieren.

Um ein Höchstmaß an Interoperabilität zu gewährleisten, setzt TextGrid weitestmöglich auf offene Standards: TEI¹⁰ und die XML-Familie¹¹, RDF¹², SOAP¹³, WSDL¹⁴, GSI¹⁵, WSRF¹⁶, SAML¹⁷ und LDAP¹⁸.

GUI

Die grafische Benutzeroberfläche von *TextGridLab* basiert auf Eclipse¹⁹ und dessen *Rich Client Platform* genanntem Framework zur Entwicklung grafischer Oberflächen. Die Bestandteile (etwa spezifische Bedienelemente für die einzelnen Tools) werden in einzelnen Modulen, sog. *Plugins* realisiert, die über gemeinsame Schnittstellen miteinander kommunizieren. Aufgrund der flexiblen *Plugin*-Architektur lässt sich die GUI individuellen Anforderungen gemäß erweitern (z.B. um Schnittstellen für zusätzliche Werkzeuge) und konfigurieren. Die Anwendung ist javabasiert und deshalb auf allen gängigen Betriebssystemen (Windows, Linux, Mac OS X) lauffähig.

Tools / Services

TextGrid setzt auf eine Service-orientierte Architektur (*Service Oriented Architecture*, SOA), ein modulares System verteilter, plattformunabhängiger *Open Source*-Komponenten, die als Web Services²⁰ angesprochen werden. Vorhandene Programme lassen sich relativ einfach in die TextGrid-Infrastruktur einbinden, indem sie als Web Services gekapselt werden – für die meisten Programmiersprachen existieren mittlerweile entsprechende Bibliotheken. Der Zugriff erfolgt über Internet-Protokolle²¹, so dass es unerheblich ist, auf welchem Server und auf welcher Plattform (Windows, Linux etc.) die Services vorgehalten werden. Lediglich Adresse und Parameter müssen bekannt sein.

¹⁰ Text Encoding Initiative, <http://www.tei-c.org>.

¹¹ Für einen kurzen Überblick siehe <http://www.w3.org/Talks/2002/ij-italy/slide11-0.html>.

¹² Resource Description Framework, <http://www.w3.org/RDF/>.

¹³ XML Protocol, <http://www.w3.org/2000/xp/Group/>.

¹⁴ Web Services Description Language, <http://www.w3.org/2002/ws/desc/>.

¹⁵ Grid Security Infrastructure, <http://www.globus.org/security/overview.html>.

¹⁶ Web Services Resource Framework, <http://www.oasis-open.org/committees/wsrf/>.

¹⁷ Security Assertion Markup Language, <http://www.oasis-open.org/committees/security/>.

¹⁸ Lightweight Directory Access Protocol, <http://www.openldap.org/>.

¹⁹ Eine Java-basierte Open Source-Entwicklungsumgebung, <http://www.eclipse.org>.

²⁰ Siehe hierzu <http://www.w3.org/2002/ws/>.

²¹ Wie SOAP (<http://www.w3.org/2000/xp/Group/>) und REST, das ausschließlich mit dem Vokabular des HTTP-Standards auskommt.

Middleware und Grid-Anbindung

Die TextGrid-Middleware bildet die Zwischenschicht zwischen den TextGrid-spezifischen Anwendungen, dem Grid und externen Datenquellen. Die Schnittstelle zum *TextGridLab* wird durch Utilities zur Datenverwaltung, Suche, Autorisierung / Authentifizierung²² und Logging modelliert. Für Recherche und Text-Retrieval werden XML-Datenbanken²³ eingesetzt, in denen Meta- und Strukturdaten redundant (d.h. zusätzlich zur Ablage im Grid-Dateisystem) gespeichert werden – und so auch über XQueries²⁴ ansprechbar sind. Für die semantische Suche (Objekt-Relationen) ist eine RDF-Datenbank²⁵ implementiert.

²² Die Autorisierung für das Lesen und Schreiben von Objekten erfolgt über ein feingranulares rollenbasiertes Zugriffskontrollsystem (RBAC, <http://csrc.nist.gov/groups/SNS/rbac/>).

²³ Derzeit im Einsatz: eXist (<http://exist.sourceforge.net>).

²⁴ W3C-Standard zur Abfrage von XML-Datenbanken, <http://www.w3.org/TR/xquery/>.

²⁵ Derzeit im Einsatz: Sesame (<http://www.openrdf.org>).